

英書 25 冊の第 43 次エントロピー計算

横 井 右 門

キーワード

エントロピー（平均情報量） entropy

冗長度 redundancy

記号 symbols

擬似コード pseudo code

1. はじめに

1957 年，ソ連のヤグロム兄弟は「情報理論入門」で，まず次のように述べている。⁽¹⁾「また他の国語についても，同じように計算できる。たとえば，ラテン文字のアルファベットには，全部で 26 コの文字がある。したがって，ラテン文字のアルファベットで書かれた文章の文字に含まれる最大の情報 ε_0 は，10 進単位で $\log 26 \doteq 1.415$ である。さらに，いろいろな国語で，それぞれの文字が現われる相対頻度を計算すると，一つの文字に含まれる平均情報量 ε_1 は，英文では 1.242（10 進）単位，仏文では 1.200，独文では 1.233 になる。これらの値はみな，1.415 より小さいことがわかる。国語によって，それぞれの文字の現われる頻度はちがうから， ε_1 の値は少しずつちがっている。」

さらに次のように述べている。⁽²⁾

「シャノンは、二つ及び三つの文字のいろいろな組合せの相対頻度の表を作って、英語での ε_2 と ε_3 を計算し、10進単位で $\varepsilon_2 = 1.075$, $\varepsilon_3 = 0.993$ であることがわかった。(前に述べたように、英語では個々の文字を独立と考えて、それぞれの文字の相対頻度だけを使って計算すれば、一つの文字に含まれる報情(ママ)は $\varepsilon_1 = 1.242$ 単位であった。さらに、英文に現われるいろいろな単語の頻度に関する統計資料を使って、シャノンは近似計算で、 ε_8 (つまり、すぐ前にある7コの文字がわかつたものとして、文章の文字一つあたりに分配される情報)がほぼ、0.6単位になることを示した。なお文章でいろいろな単語がどのようにつづいて配列されねばならないかという規則(collocation)をしらべたり、多くの単語の組合せを統計的にくわしく計算したりすると、みかけ上、情報ははるかに減るように思われるが、実際には0.6より大して減らない。

このように意味がある文章の文字一つあたりに平均的に分配される情報は、アルファベットの文字に含まれている最大情報の半分以下になってしまうことがわかる。(英語では、 $\varepsilon_0 = \log 26 \doteq 1.415$ 単位で、 $\varepsilon_8 = 0.6$ 単位である)。ある国語で、意味をもっている文章の一つの文字に平均的に分配される情報とアルファベットの文字に関する最大の情報量との比を1から引いたもの(つまり、式では、 $1 - \frac{\varepsilon_\infty}{\varepsilon_0} = 1 - \frac{\varepsilon_\infty}{\log n}$)をシャノン流に、その国語の冗長度と名づければ、よく普及しているヨーロッパの言語の冗長度は、50% (或いはそれ以上)である。いいかえれば、通常のどんな文章でも、50%以上の文字は、言語構造そのものによって、いつも必然的に決ってしまう。」

このヤグロムの記述で「単位」で終わっている数値は、常用対数による表現であり、これらを2進単位の大数に変換すれば、後述するバーナードのエントロピー計算値にすべて合致してしまう。何故、常用対数にしたのか、バーナードの計算をシャノンの計算としたのか不可解である。

堀は次のように述べている。⁽³⁾

「何しろ、前にみたように、文字のつながり方の制約は、三つや四つどころでなく、もっともっと長い文字の連鎖まで及ぶことが明らかなのだから。た

ぶん、少なくとも十四か十五の相つづく文字の列のすべての可能なものを数えあげ、それらの頻度分布をしらべてエントロピーを計算しなければ、実際の値に近い値は求まらないだろう。しかしそれは大変な仕事で、まだ誰もやっていない。ある人が英語の中に出てくる相つづく八つの文字の出現頻度をしらべてエントロピーを計算したのが今までの最高記録である。その値は二・三五であったが、実際はこれよりもっともっと小さいだろう。あとは大ざっぱに見当をつけるほかないのだが、まず一・三ぐらいだろうというのが、今のところの定説になっている」

この二・三五というのもバーナードの計算である。ヤグロムにしても、堀にしてもバーナードの計算に頼っている。

そこで改めて相つづく 43 字までの出現頻度からエントロピーを計算し直すというのが本稿の目的である。

2. バーナードのエントロピー計算

南は G.A.Barnard が 1955 年に Statistical Calculation of Word Entropies for Four European Languages, IRE Trans-IT-1 において次のように計算値を発表していると紹介している。⁽⁴⁾

	英語	仏語	独語	スペイン語
1 次	4.124	3.98	4.10	4.015
2 次	3.56			
3 次	3.3			
8 次	2.35			

ヤグロムの 1.242, 1.200, 1.233 を 2 進単位の対数に換算すると 4.126, 3.986, 4.096 となり、バーナードの 1 次エントロピーに極めて近い値になる。ヤグロムがシャノンの計算としている 1.076, 0.993 は 2 進単位に換算すると 3.574, 3.299 になる。20 インチ計算尺の視読誤差の範囲に収まる。ヤグロムは ε 8 が 0.6 単位であるとしているが、2 進単位の 2.35 を 10 進単位に換算すると

0.707 になるが、小数部第一位で植字の段階で間違えて、0.6 になったと想像することは可能である。

バーナードの計算は藤田も引用している。⁽⁵⁾

南が積極的にバーナードの名前を出している姿勢は高く評価しなければならない。

3. エントロピー

計算するのは熱力学のエントロピーではなく、情報理論のエントロピーである。単位はビット数である。熱力学のエントロピーの単位がビット数であるわけがない。

前川によればエントロピーの定義はつぎのとおりである。⁽⁶⁾

「言語が n 個の記号（番号 1 から n で区別する）で表現され、それぞれの出現確率が

$P(n)$ であるとき、つぎの式で定義される。

$$\text{エントロピー} = \sum_{i=1}^n P(i) \log_2 P(i)$$

ただし、この値は負になるので、符号を逆転させる。このエントロピーが重要なのは、この値は実は、分析の対象である言語において、1 つの記号を送るのに必要な平均ビット数を表わしているからである。」

シャノンではアルファベットとスペースを使ったから、英語の 1 次エントロピーの計算では 27 個の記号を使ったことになる。英語の 2 次エントロピーを計算するためには 27^2 すなわち 729 個の記号を使えばよい。英語の 3 次エントロピーを計算するためには 27^3 すなわち 19,683 個の記号を使うことになる。16 次エントロピーは 27^{16} 個の記号を使うことになる。

しかし後述するデータの 2 番目 The Hunt for Red October について言えば

実際に発生した 2 連字は論理的に可能な 729 に対し 609 個である。3 連字の論理的可能性は 19,683 個であるが、実際に発生するのは 5,877 個である。29 パーセントにすぎない。配列を使って計算するのは資源の無駄遣いである。それよりは n 次エントロピーを計算するとき n 字の文字列レコードを sort し (整列させ) たほうが得策である。

4. データ

次の 25 冊である。全て手作業入力でテキスト・データにしてある。

1. イギリス伝承童謡 : Who Killed Cock Robin
2. Tom Clancy: The Hunt for Red October, Harper Collins, 1993
3. Tom Clancy: OP Center, Berkley Novels, 1995
4. Tom Clancy: Without Remorse, Berkley Novels, 1993
5. Frederick Forsyth: The Day of Jackal, Bantam Books, 1993
6. Frederick Forsyth: The Dogs of War, Bantam Books, 1995
7. Frederick Forsyth: The Odessa File, Bantam Books, 1993
8. Frederick Forsyth: The Shepherd, Corgi Books, 1990
9. Jostein Gaarder: Sophie's World, Berkley Books, 1996
10. Akira Kohchi: Why I survived A-Bomb,
Institute for Historical Review, 1989
11. Anne McCaffrey: The Crystal Singer, Corgi Books, 1991
12. Anne McCaffrey: Crystal Line, Del Rey Books, 1992
13. Robert B. Parker: A Catskill Eagle, Dell, 1985
14. Robert B. Parker: Pastime, Berkley Novel, 1992
15. Robert B. Parker: God Save the Child, Penguin Books, 1997
16. Robert B. Parker: The Godwulf Manuscript, Dell, 1987
17. A. J. Quinnell: Man on Fire, Orion, 1994
18. A. J. Quinnell: In the Name of the Father, Signet, 1987

19. A. J. Quinnell: The Blue Ring, Orion, 1994
20. A. J. Quinnell: Message from Hell, Orion, 1996
21. H. G. Wells: A Short History of the World, Collins, 1953
22. J. K. Rowling: Harry Potter and the Sorcerer's Stone,
Bloomsbury Publishing Plc, 1997
23. C. S. Lewis: Prince Caspian, Collins, 1980
24. Yoshimoto Banana: N.P., Grove Press, 1994
25. Arthur C. Clarke: A Space Odyssey, A ROC Book, 1968

後で必要になるので一番目のデータの全文を次に紹介しておく。

Who killed Cock Robin?

I, said the Sparrow,
With my bow and arrow,
I killed Cock Robin.

Who saw him die?

I, said the Fly,
With my little eye,
I saw him die.

Who caught his blood?

I, said the Fish,
With my little dish,
I caught his blood.

Who'll make the shroud?

I, said the Beetle,
With my thread and needle,

I'll make the shroud.

Who'll dig his grave?

I, said the Owl,

With my pick and shovel,

I'll dig his grave.

Who'll be the parson?

I, said the Rook,

With my little book,

I'll be the parson.

Who'll be the clerk?

I, said the Lark,

If it's not int the dark,

I'll be the clerk.

Who'll carry the link?

I, said the Linnet,

I'll fetch it in a minute,

I'll carry the link.

Who'll be chief mourner?

I, said the Dove,

I mourn for my love,

I'll be chief mourner.

Who'll carry the coffin?

I, said the Kite,
If it's not through the night,
I'll carry the coffin.

Who'll bear the pall?
We, said the Wren,
Both the cock and the hen,
We'll bear the pall.

Who'll sing a psalm?
I, said the Thrush,
As she sat on bush,
I'll sing a psalm.

Who'll toll the bell?
I, said the Bull,
Because I can pull,
I'll toll the bell.

All the birds of the air,
Fell a sighing and a sobbing,
When they heard the bell toll
For poor Cock Robin.

5. エントロピー計算

原始テキスト・データからエントロピーの次数に対応した文字列レコード・ファイルと総レコード数のみを記録した1レコード・ファイルを出力する。

文字列レコード・ファイルを整列させる (sort する)。整列済みファイルを 1 レコードずつ読み、同一文字列のレコード数を数え、それを総レコード数で割り、出現相対頻度を求め、エントロピーを計算していく。この 3 ステップからなる。

前川に倣い⁽⁷⁾、擬似コード pseudo coding で説明する。第 40 次エントロピー計算を例にして説明する。

第 1 ステップ

```
#include <stdio.h>
```

```
#include <string.h>
```

```
void main(void)
```

```
{
```

```
    変数, 配列の定義
```

```
    ファイルのオープン処理
```

```
    先頭の 1 文字を読む
```

```
    次の手続きを 40 回反復する
```

```
    {
```

```
        大文字ならば小文字にする。
```

```
        改行キーだったらスペースにする
```

```
        文字配列の適当な要素に順次, 文字を書きこむ
```

```
        次の 1 文字を読む
```

```
    }
```

```
    while( ファイルが終わるまで )
```

```
    {
```

```
        大文字ならば小文字にする
```

```
        文字が改行キーだったらスペースにする
```

```
        文字配列の 40 番目の要素に文字を入れる
```

```
        文字配列の 41 番目の要素に pure binary zero を入れる
```

40 連字の文字列レコードを出力する。

改行キーを出力する (1 レコードの出力が完了する)

文字配列の各要素を一文字ずつ前にずらす

次の一文字を読む

}

統計 1 レコードファイルを出力する

全ファイルをクローズする

}

具体的にプログラム Sigi40.c を次に示す。

```
001:/* Sigi40.c output 40-byte strings */
002:#include <stdio.h>
003:#include <string.h>
004:void main(void)
005:{
006:    int i = 0, stringcnt = 0;
007:    char chara, infile[50], work[50], outfile[50];
008:    FILE * fp0, * fp1, * fp3;
009:    printf("Input-file name: ");
010:    scanf("%s", infile);
011:    printf("Output-file name: ");
012:    scanf("%s", outfile);
013:    fp0 = fopen(infile, "r");
014:    fp1 = fopen(outfile, "w");
015:    fp3 = fopen("filesize.txt", "w");
016:    chara = getc(fp0);
017:    while (i < 39)
018:    {
019:        if(65 <= chara && chara <= 90)
020:        {
021:            chara = chara + 32;
022:        }
023:        if(chara == 0x0a)
024:        {
025:            chara = 32;
026:        }
027:        if((97 <= chara && chara <= 122) || (chara == 32))
028:        {
029:            work[i] = chara;
030:            i = i + 1;
031:        }
032:        chara = getc(fp0);
```

```

033:     }
034:     if(fp0 != NULL)
035:     {
036:         while(chara != EOF)
037:         {
038:             if(65 <= chara && chara <= 90)
039:             {
040:                 chara = chara + 32;
041:             }
042:             if(chara == 0x0a)
043:             {
044:                 chara = 32;
045:             }
046:             if((97 <= chara && chara <= 122) || (chara == 32))
047:             {
048:                 work[39] = chara;
049:                 work[40] = '\0';
050:                 fputs(work, fp1);
051:                 fputc('\n', fp1);
052:                 stringcnt = stringcnt + 1;
053:                 for(i = 0; i < 39; i++)
054:                 {
055:                     work[i] = work[i + 1];
056:                 }
057:             }
058:             chara = getc(fp0);
059:         }
060:         fprintf(fp3, "%d\n", stringcnt);
061:     }
062:     fclose(fp0);
063:     fclose(fp1);
064:     printf("%d strings \n", stringcnt);
065:     fclose(fp3);
066: }

```

第 2 ステップ

文字列レコードファイルを整列させる (sort する)

第 3 ステップ

```

#include <stdio.h>

#include <string.h>

#include <math.h>

```

```

void main(void)
{
    変数, 配列の定義
    ファイルのオープン処理
    if( 入力ファイルがオープンできなければ )
    {
        何もしない
    }
    else
    {
        統計ファイル (1 レコード・ファイル) を読んで
        文字列数を知る
        先頭の文字列を読む
        その文字列を trace に入れる
        while( ファイルが終わるまで )
        {
            if( 文字列が trace に等しければ )
            {
                文字列カウンターを増分する
            }
            else
            {
                文字列カウンターの値を文字列数で割って確率を求める
                確率 × log( 確率 ) を entropy に加算する
                文字列を trace に入れる
                文字列カウンターを 1 にする
                unique カウンターを増分する
            }
        }
    }
}

```

次の文字列レコードを読む

}

}

ファイルをクローズする

entropy を 40 で割る

画面に 40 次エントロピーを表示する

}

次にプログラム Miller40.c を示す

```
001:/* Miller40.c Calculation of 40th order entropy */
002:#include <stdio.h>
003:#include <string.h>
004:#include <math.h>
005:
006:void main(void)
007:{
008:    double entropy = 0, proba = 0;
009:    int totalcnt, stringcnt = 0, uniquecnt = 1;
010:    char trace[50], infile[50], work[50];
011:    const char * miller;
012:    FILE * fp1, *fp3;
013:    printf("Input-file name: ");
014:    scanf("%s", infile);
015:    fp1 = fopen(infile, "r");
016:    fp3 = fopen("filesize.txt", "r");
017:    if(fp1 == NULL)
018:    {
019:    }
020:    else
021:    {
022:        fscanf(fp3, "%d", &totalcnt);
023:        printf("%7d\n", totalcnt);
024:        miller = fgets(trace, 50, fp1);
025:        strcpy(work, trace);
026:        while(miller != NULL)
027:        {
028:            if(strcmp(trace, work) == 0)
029:            {
030:                stringcnt = stringcnt + 1;
031:            }
032:            else
033:            {
034:                proba = (float)stringcnt / (float)totalcnt;
```

```

035:          entropy = entropy + proba * log(proba) / log(2);
036:          uniquecnt = uniquecnt + 1;
037:          strcpy(trace, work);
038:          stringcnt = 1;
039:      }
040:      miller = fgets(work, 50, fp1);
041:  }
042:  proba = (float)stringcnt / (float)totalcnt;
043:  entropy = entropy + proba * log(proba) / log(2);
044:  }
045:  fclose(fp1);
046:  fclose(fp3);
047:  entropy = entropy / 40.0;
048:  printf("(Miller40.c) ");
049:  printf("%d unique strings\n", uniquecnt);
050:  printf(" 40th order entropy = %8.6f\n", entropy);
051:}

```

このプログラム第47行の除数を変更するだけで何次エントロピーでも計算できる

6. 計算結果

書 名	Who Killed Cock Robin	The Hunt for Red October	OP Center	Without Remorse	The Day of Jackal
file size	1,289	970,567	537,312	1,462,084	771,683
H01	3.951722	4.080464	4.058987	4.076145	4.063707
H02	3.306571	3.743244	3.710725	3.720784	3.695529
H03	2.618509	3.411523	3.380817	3.379858	3.361577
H04	2.118716	3.092126	3.067087	3.069802	3.053169
H05	1.769667	2.794259	2.794259	2.811472	2.784862
H06	1.523971	2.587464	2.554308	2.593109	2.551317
H07	1.333189	2.380843	2.339259	2.400077	2.344505
H08	1.184097	2.194507	2.145373	2.224664	2.158803
H09	1.066903	2.024234	1.969411	2.062322	1.990291
H10	0.970298	1.869798	1.811584	1.912473	1.838261
H11	0.890077	1.730608	1.671115	1.775313	1.710752
H12	0.822563	1.605729	1.546549	1.650486	1.579231
H13	0.764931	1.494324	1.436555	1.537954	1.469920
H14	0.713608	1.395297	1.339477	1.437128	1.372680
H15	0.668253	1.307220	1.253662	1.346904	1.286127
H16	0.628347	1.228752	1.177527	1.266215	1.208916
H17	0.592928	1.158621	1.109682	1.193932	1.139917
H18	0.561011	1.095684	1.048941	1.128974	1.078012
H19	0.532077	1.038983	0.994326	1.070424	1.022239
H20	0.505767	0.987696	0.945012	1.017461	0.971793
H21	0.481793	0.941116	0.900278	0.969337	0.925976
H22	0.459834	0.898650	0.859537	0.925555	0.884204
H23	0.439785	0.859796	0.822296	0.885472	0.845988
H24	0.421406	0.824123	0.788129	0.848683	0.810899
H25	0.404498	0.791267	0.756676	0.814806	0.778574
H26	0.388900	0.760912	0.727625	0.783514	0.748709
H27	0.374438	0.732787	0.700717	0.764528	0.721035
H28	0.360972	0.706658	0.675722	0.727603	0.695324
H29	0.348625	0.682323	0.652446	0.702530	0.671377
H30	0.336864	0.659603	0.630717	0.679124	0.649019
H31	0.325955	0.638345	0.610387	0.657226	0.628100
H32	0.315728	0.618411	0.591324	0.636694	0.608485
H33	0.306120	0.599682	0.573416	0.617405	0.590056
H34	0.297078	0.582053	0.556560	0.599251	0.572709
H35	0.288553	0.565430	0.540666	0.582132	0.556351
H36	0.280501	0.549730	0.525653	0.565965	0.450901
H37	0.272884	0.534877	0.511452	0.550671	0.526285
H38	0.265668	0.520805	0.497996	0.536181	0.512438
H39	0.258823	0.507454	0.485231	0.522435	0.499300
H40	0.252319	0.494770	0.473104	0.509375	0.486810
H41	0.246133	0.482705	0.461568	0.496952	0.474946
H42	0.240241	0.471213	0.450580	0.485120	0.468638
H43	0.236623	0.460256	0.440104	0.473839	0.452857

書 名	The Dogs of War	The Odessa File	The Shepherd	Sophie's World	Why I survived A-Bombs
file size	810,586	596,877	64,003	964,734	371,522
H01	4.068114	4.063637	4.077524	4.054536	4.105486
H02	3.704562	3.705517	3.714431	3.700202	3.752481
H03	3.374857	3.369982	3.341634	3.362045	3.436056
H04	3.063209	3.055386	2.974577	3.045866	3.123731
H05	2.793966	2.782213	2.643293	2.776917	2.836641
H06	2.561120	2.554118	2.352480	2.546776	2.576470
H07	2.354661	2.332727	2.103107	2.344339	2.341912
H08	2.168820	2.142822	1.889770	2.162480	2.132168
H09	1.999820	1.970830	1.708196	1.997151	1.945393
H10	1.847139	1.816338	1.554171	1.847501	1.781172
H11	1.710030	1.678350	1.422606	1.712793	1.637667
H12	1.586969	1.555304	1.309746	1.591830	1.512284
H13	1.477710	1.446053	1.212490	1.483636	1.402503
H14	1.379265	1.349292	1.128153	1.387000	1.306310
H15	1.292158	1.263434	1.054395	1.300752	1.221707
H16	1.214455	1.187052	0.989399	1.223658	1.146975
H17	1.144988	1.118882	0.931779	1.154542	1.080553
H18	1.082655	1.057814	0.880417	1.092377	1.021236
H19	1.026520	1.002866	0.834372	1.036254	0.967982
H20	0.975753	0.953210	0.793413	0.985426	0.919940
H21	0.929659	0.908151	0.755257	0.939213	0.876376
H22	0.887654	0.867103	0.721016	0.897035	0.836707
H23	0.849232	0.829560	0.689735	0.858409	0.800444
H24	0.813962	0.795106	0.661045	0.822920	0.767172
H25	0.781482	0.763378	0.634638	0.790211	0.736542
H26	0.751480	0.734074	0.610251	0.759972	0.708258
H27	0.723685	0.706927	0.587662	0.731943	0.682057
H28	0.697864	0.681708	0.566680	0.705893	0.657719
H29	0.673818	0.658222	0.547146	0.681623	0.635054
H30	0.651370	0.636298	0.528913	0.658959	0.613899
H31	0.630368	0.615784	0.511854	0.637749	0.594105
H32	0.610677	0.596550	0.495860	0.617857	0.575547
H33	0.592177	0.578480	0.480834	0.599165	0.558111
H34	0.574763	0.561472	0.466692	0.581569	0.541700
H35	0.558345	0.545436	0.453358	0.564975	0.526225
H36	0.542837	0.530289	0.440765	0.549300	0.511610
H37	0.528168	0.515960	0.428853	0.534471	0.497785
H38	0.514270	0.502384	0.417568	0.520421	0.484686
H39	0.501085	0.489505	0.406861	0.507090	0.472260
H40	0.488559	0.477268	0.396690	0.494424	0.460454
H41	0.476644	0.465629	0.387014	0.482376	0.449224
H42	0.465296	0.454544	0.377800	0.470900	0.438530
H43	0.454475	0.443974	0.369014	0.459957	0.428332

英書 25 冊の第 43 次エントロピー計算

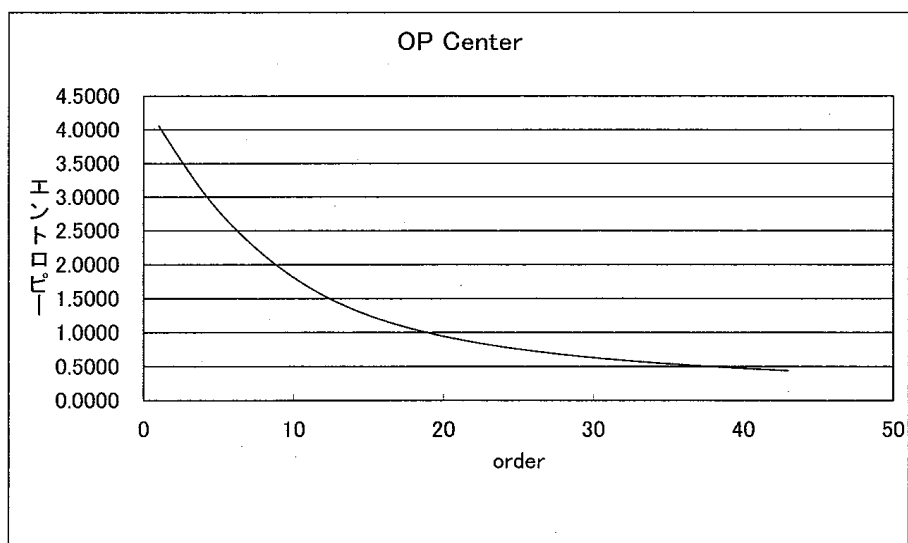
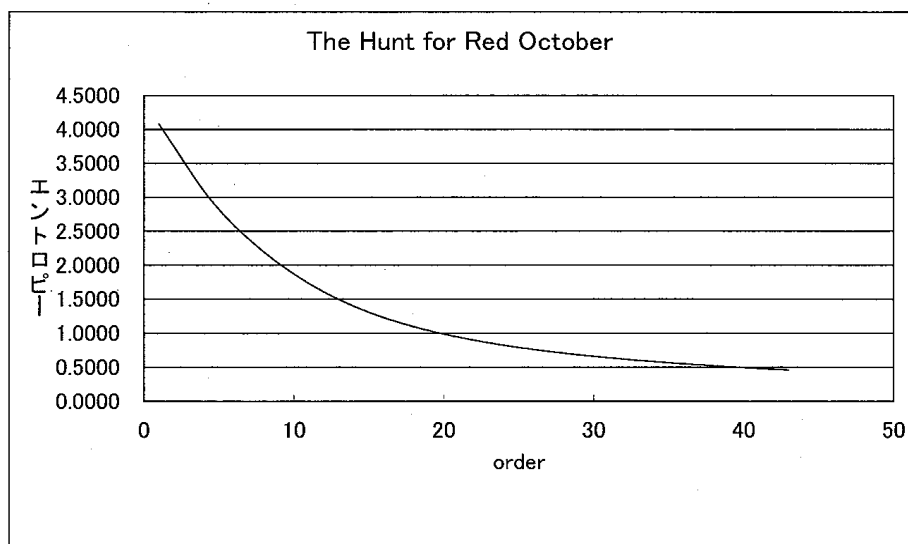
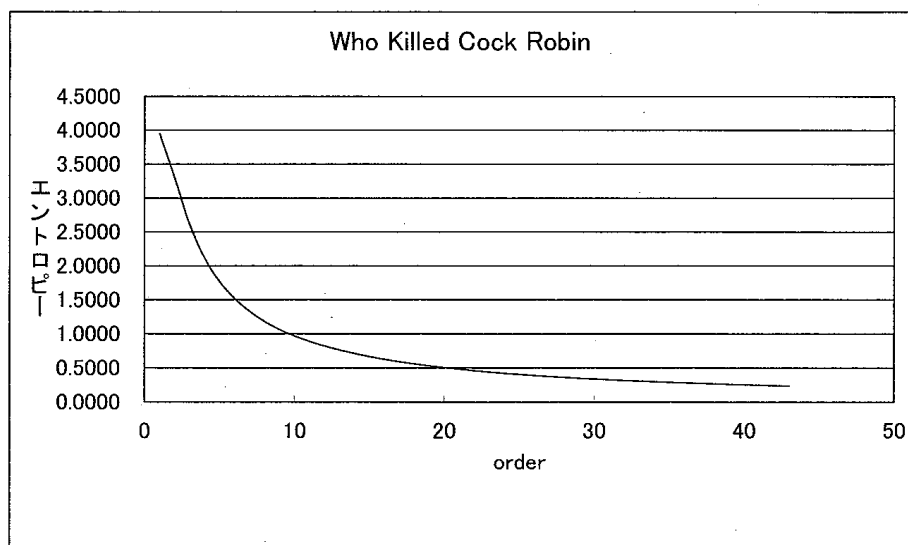
書 名	The Crystal Singer	Crystal Line	A Catskill Eagle	Pastime	God Save the Child
file size	304,275	511,075	404,989	287,287	298,069
H01	4.090534	4.071662	4.028679	4.036607	4.053653
H02	3.725994	3.715240	3.674846	3.682574	3.700939
H03	3.385547	3.372888	3.319868	3.321185	3.348583
H04	3.066058	3.053926	2.999099	2.993473	3.023943
H05	2.792212	2.781191	2.726949	2.713752	2.742900
H06	2.555564	2.542879	2.485689	2.467347	2.492648
H07	2.345113	2.330058	2.271828	2.249011	2.269617
H08	2.155192	2.137730	2.080731	2.054667	2.071097
H09	1.982324	1.962598	1.909356	1.880796	1.893797
H10	1.826648	1.805301	1.757230	1.727035	1.737220
H11	1.687490	1.665400	1.622684	1.591465	1.599622
H12	1.563423	1.541292	1.503675	1.472227	1.478810
H13	1.453470	1.431649	1.398474	1.367312	1.372829
H14	1.356229	1.334961	1.305367	1.274891	1.279663
H15	1.270137	1.249479	1.222718	1.193232	1.192516
H16	1.193644	1.171389	1.149107	1.120759	1.124714
H17	1.125346	1.106084	1.083324	1.056129	1.059000
H18	1.064086	1.045547	1.024341	0.998278	1.001922
H19	1.008912	0.991111	0.971236	0.946257	0.949776
H20	0.959006	0.941931	0.923222	0.899282	0.901677
H21	0.913687	0.897310	0.879641	0.856671	0.859963
H22	0.872385	0.856667	0.839925	0.817870	0.821061
H23	0.834616	0.819512	0.803596	0.782404	0.785489
H24	0.799941	0.785426	0.770245	0.749872	0.752845
H25	0.768014	0.754048	0.739527	0.719925	0.722787
H26	0.738522	0.725070	0.711150	0.692269	0.695024
H27	0.711202	0.698233	0.684858	0.666652	0.669308
H28	0.685828	0.673309	0.660432	0.642862	0.645423
H29	0.662198	0.650100	0.637684	0.620708	0.623184
H30	0.640140	0.628437	0.616446	0.600027	0.602424
H31	0.619501	0.608169	0.596572	0.580800	0.583001
H32	0.600150	0.589168	0.577939	0.562539	0.564791
H33	0.581971	0.571318	0.560433	0.545497	0.547683
H34	0.564859	0.554517	0.543955	0.529456	0.531579
H35	0.548724	0.538675	0.528418	0.514331	0.516395
H36	0.533484	0.523713	0.513743	0.500045	0.502053
H37	0.519067	0.509560	0.499861	0.486531	0.488487
H38	0.505409	0.496152	0.486709	0.473728	0.475634
H39	0.492451	0.483431	0.474230	0.461581	0.463440
H40	0.480141	0.471346	0.462338	0.450041	0.451855
H41	0.468431	0.459850	0.451099	0.439065	0.440834
H42	0.457278	0.448902	0.440359	0.428611	0.430339
H43	0.446644	0.438463	0.430119	0.418643	0.420332

書 名	The Godwulf Manuscript	Man on Fire	In the Name of the Father	The Blue Ring	Message from Hell
file size	304,275	547,589	638,157	622,674	298,069
H01	4.054947	4.063091	4.073119	4.060667	4.059951
H02	3.700538	3.704640	3.713228	3.701047	3.697347
H03	3.346448	3.361731	3.375556	3.356503	3.347897
H04	3.021676	3.045617	3.059764	3.035953	3.022360
H05	2.741941	2.772321	2.786789	2.760488	2.743293
H06	2.494004	2.534545	2.549188	2.522708	2.501684
H07	2.272396	2.324328	2.338639	2.314341	2.289669
H08	2.076223	2.135529	2.149486	2.128117	2.100531
H09	1.899216	1.963959	1.977912	1.959533	1.930142
H10	1.742477	1.809604	1.823345	1.808246	1.778225
H11	1.604327	1.671622	1.684871	1.672941	1.643068
H12	1.482858	1.548532	1.561571	1.552115	1.522970
H13	1.376322	1.439195	1.452132	1.444519	1.416383
H14	1.282688	1.342355	1.355186	1.349028	1.322076
H15	1.200156	1.256520	1.269155	1.264136	1.238465
H16	1.127064	1.180233	1.192583	1.188474	1.164032
H17	1.062024	1.112232	1.124214	1.120798	1.097520
H18	1.003849	1.051356	1.062925	1.060066	1.037871
H19	0.951544	0.996595	1.007768	1.005328	0.984138
H20	0.904327	0.947138	0.957918	0.955778	0.935524
H21	0.861515	0.902277	0.912663	0.910772	0.891379
H22	0.822535	0.861424	0.871417	0.869733	0.851142
H23	0.786903	0.824077	0.833694	0.832167	0.814329
H24	0.754213	0.789816	0.799071	0.797667	0.780537
H25	0.724116	0.758280	0.767192	0.765884	0.749417
H26	0.696311	0.729156	0.737742	0.720665	0.720665
H27	0.670554	0.702179	0.710459	0.709294	0.694023
H28	0.646630	0.677122	0.685117	0.684003	0.669272
H29	0.624350	0.653788	0.661515	0.660446	0.646220
H30	0.603551	0.632006	0.639480	0.638452	0.624698
H31	0.584092	0.611627	0.618864	0.617873	0.604561
H32	0.565847	0.592520	0.599534	0.598577	0.585679
H33	0.548707	0.574570	0.581373	0.580488	0.567939
H34	0.532573	0.557674	0.564279	0.563383	0.551242
H35	0.517361	0.541743	0.548161	0.547293	0.535497
H36	0.502992	0.526697	0.532937	0.532095	0.520627
H37	0.489400	0.512463	0.518536	0.517717	0.506558
H38	0.476522	0.498978	0.504892	0.504095	0.493230
H39	0.463440	0.486185	0.491947	0.494117	0.480585
H40	0.452697	0.474031	0.479650	0.478893	0.468572
H41	0.441656	0.462470	0.467952	0.467213	0.457144
H42	0.431141	0.451459	0.456811	0.456090	0.446260
H43	0.421114	0.440960	0.440960	0.445483	0.435882

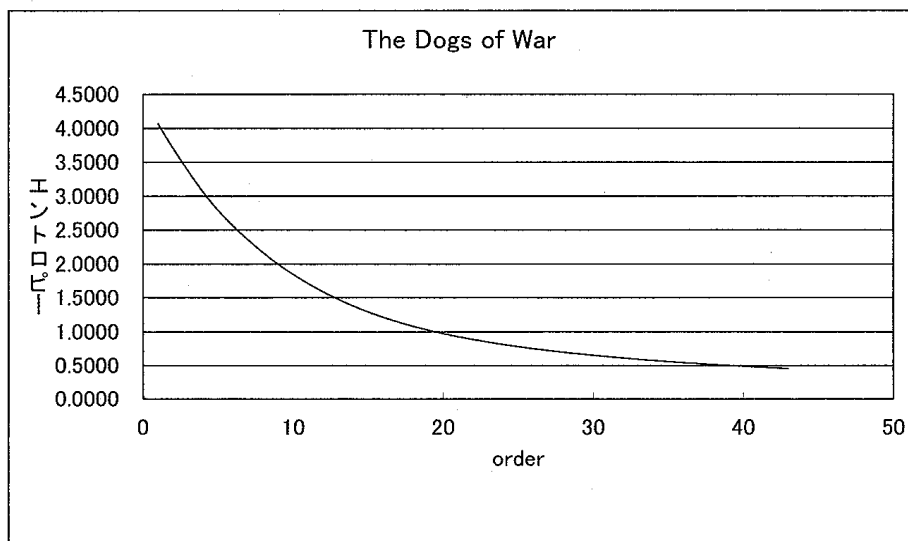
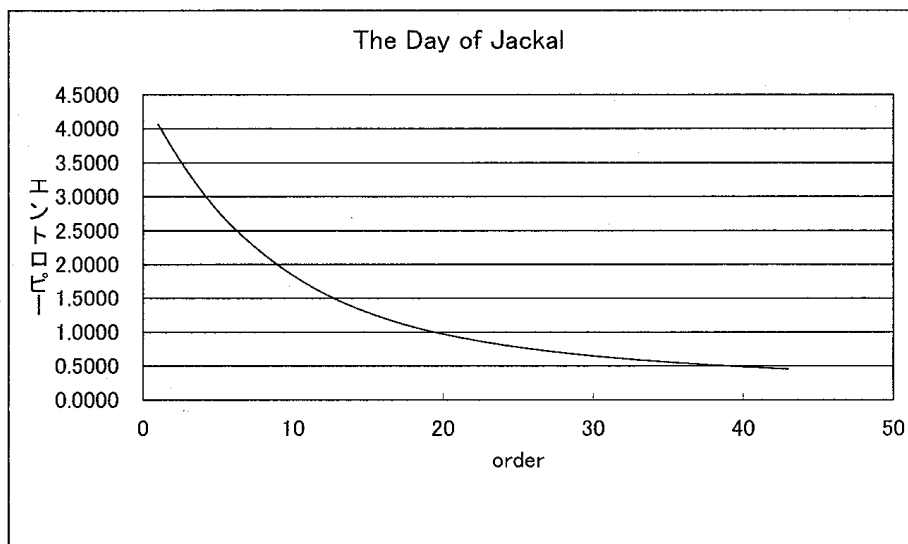
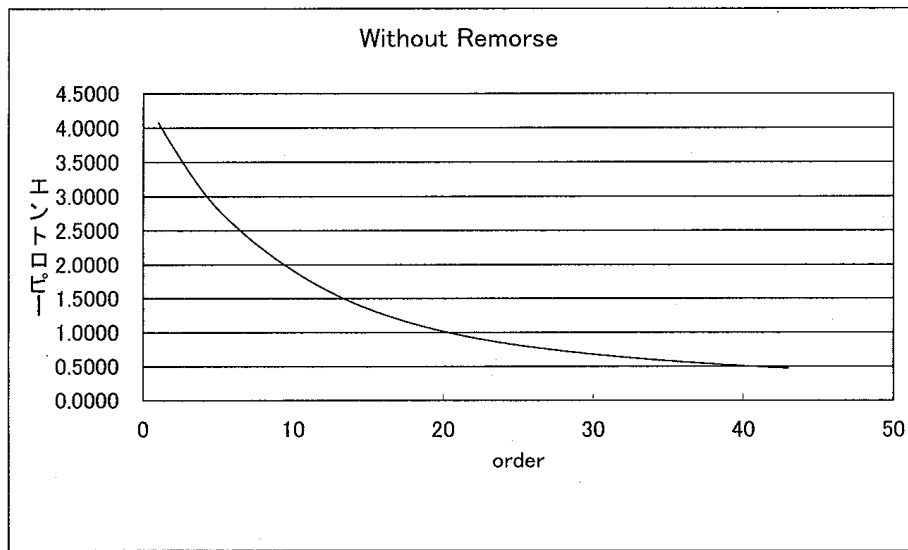
英書 25 冊の第 43 次エントロピー計算

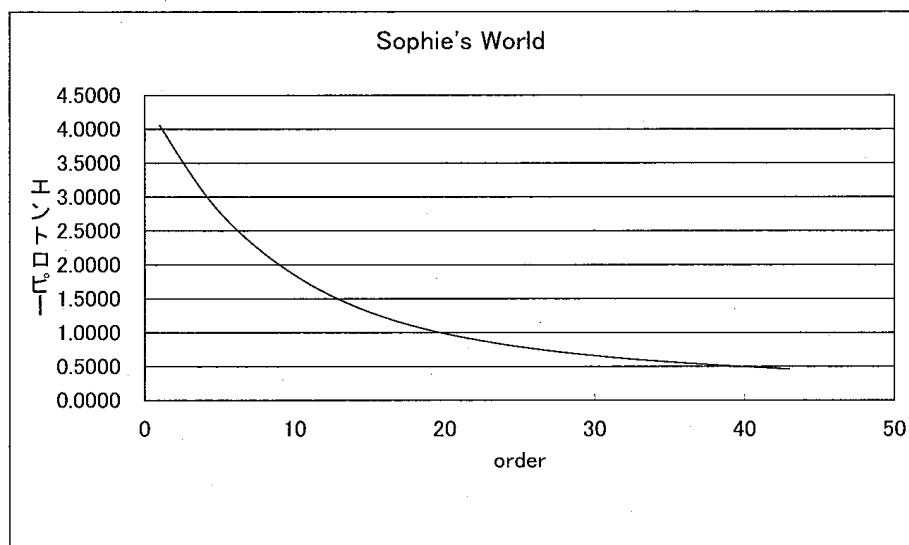
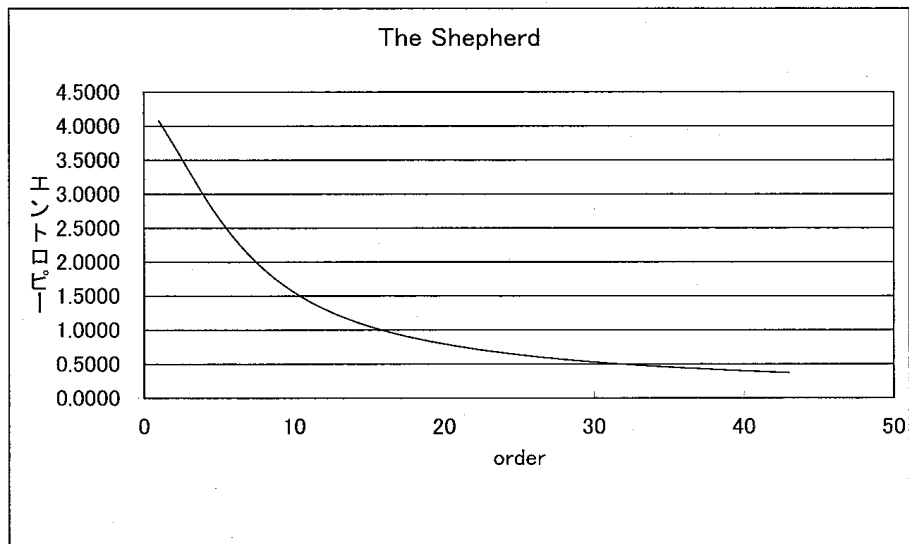
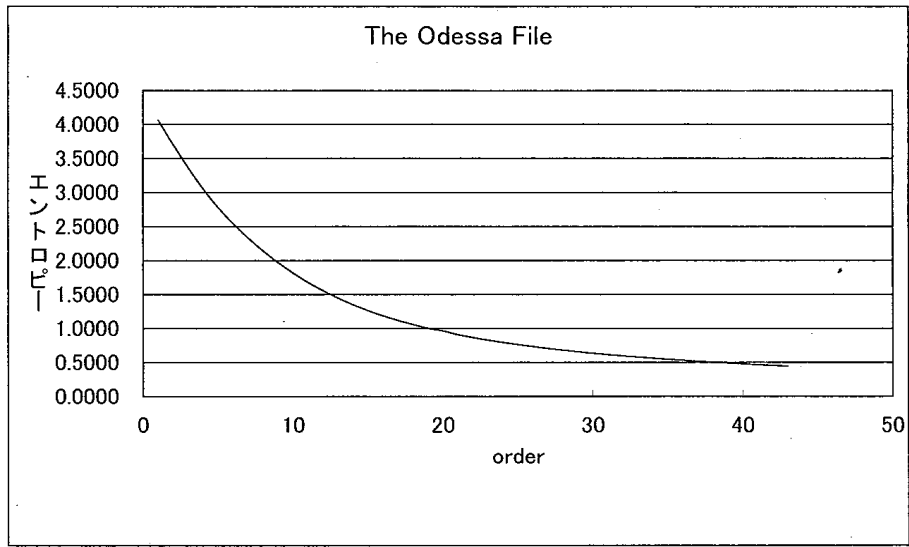
書 名	A Short History of the World	Harry Potter and the Sorcerer's	Prince Caspian	N.P.	A Space Odyssey
file size	712,476	441,674	252,542	162,125	368,733
H01	4.084961	4.086546	4.046785	4.019926	4.069920
H02	3.711772	3.718182	3.674851	3.665617	3.709878
H03	3.380499	3.353834	3.293484	3.294808	3.380259
H04	3.060751	3.022613	2.964378	2.956081	3.059985
H05	2.776146	2.745592	2.685717	2.671201	2.777302
H06	2.533825	2.504658	2.443111	2.425580	2.527429
H07	2.324854	2.290954	2.229662	2.210769	2.305317
H08	2.140635	2.099598	2.038848	2.019516	2.107084
H09	1.974377	1.927711	1.866635	1.848176	1.929109
H10	1.824370	1.774556	1.713531	1.696221	1.771135
H11	1.689697	1.638504	1.578713	1.562008	1.631776
H12	1.569066	1.518047	1.460077	1.443610	1.509064
H13	1.461264	1.411544	1.355666	1.339723	1.401036
H14	1.365063	1.317247	1.263784	1.248390	1.305797
H15	1.279253	1.233618	1.182656	1.167882	1.221635
H16	1.202698	1.159268	1.110772	1.096569	1.147051
H17	1.134131	1.092872	1.046673	1.033160	1.080666
H18	1.072568	1.033373	0.989314	0.976471	1.021322
H19	1.017101	0.979813	0.937750	0.925527	0.968030
H20	0.966938	0.931399	0.891183	0.879560	0.919934
H21	0.921346	0.887468	0.848961	0.837897	0.876331
H22	0.879815	0.847430	0.810519	0.799963	0.836644
H23	0.841803	0.810811	0.775379	0.765289	0.800359
H24	0.806900	0.777188	0.743146	0.733477	0.767071
H25	0.774748	0.746210	0.713466	0.704191	0.736432
H26	0.745039	0.717583	0.686054	0.677142	0.708137
H27	0.717508	0.691058	0.660664	0.652088	0.681931
H28	0.691929	0.666417	0.637083	0.628817	0.657592
H29	0.668102	0.643466	0.615126	0.607146	0.634930
H30	0.645824	0.622039	0.594631	0.586916	0.613776
H31	0.625038	0.601990	0.575456	0.567990	0.593984
H32	0.605518	0.583191	0.557478	0.550245	0.575428
H33	0.587179	0.565529	0.540589	0.533575	0.557996
H34	0.569915	0.548904	0.524693	0.517884	0.541588
H35	0.553638	0.533227	0.509704	0.503089	0.526117
H36	0.538264	0.518419	0.495547	0.489116	0.511506
H37	0.523720	0.504412	0.482156	0.475898	0.497684
H38	0.509941	0.491141	0.469469	0.463375	0.484590
H39	0.496868	0.478551	0.457432	0.451495	0.472167
H40	0.484448	0.466589	0.445997	0.440209	0.460364
H41	0.472634	0.455211	0.435120	0.429473	0.449138
H42	0.461383	0.444374	0.424760	0.419248	0.438445
H43	0.450654	0.434041	0.414883	0.409499	0.428250

7. グラフ

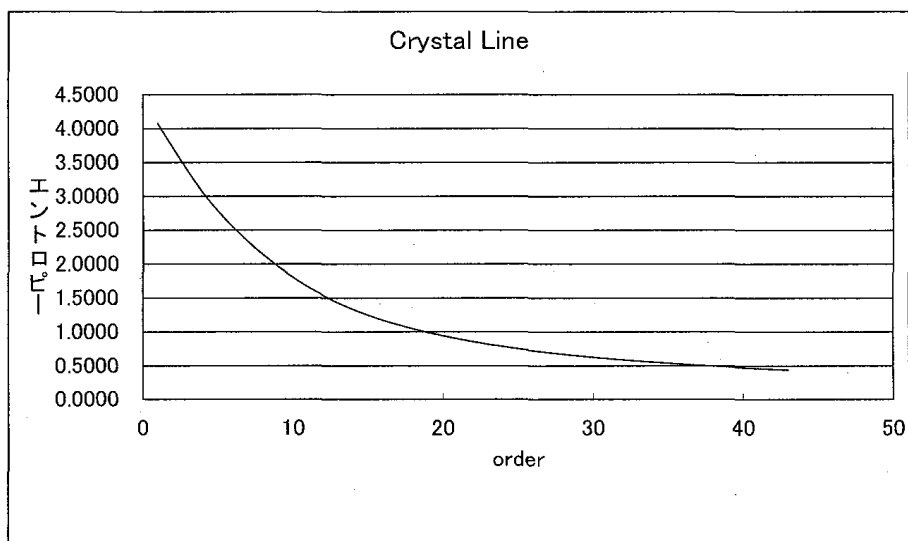
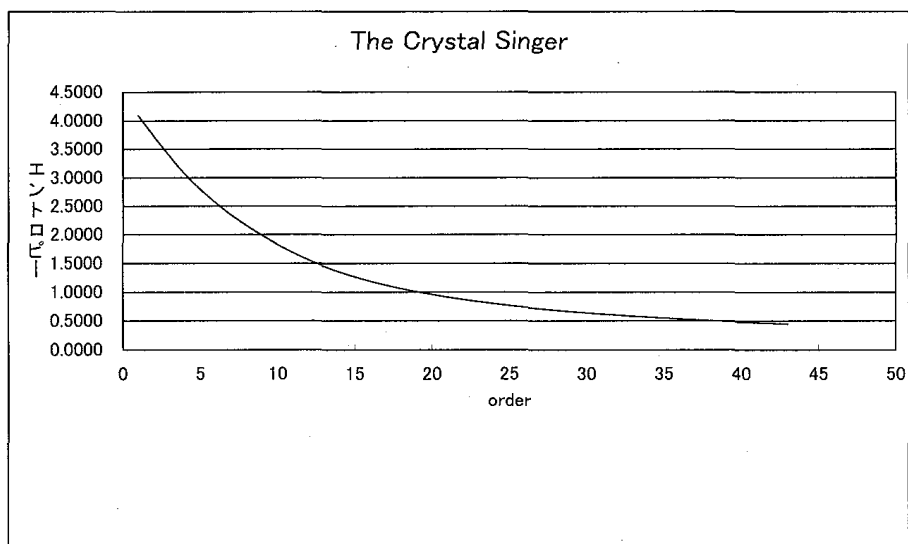
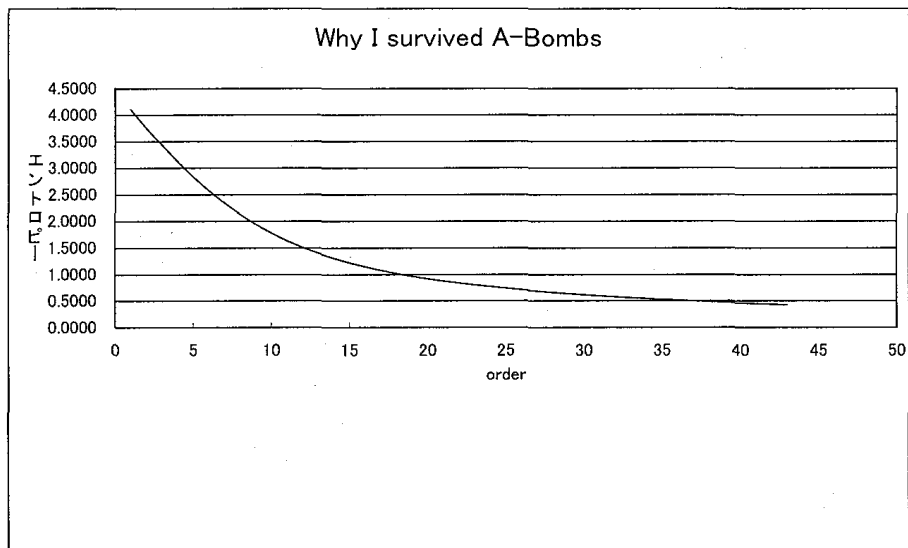


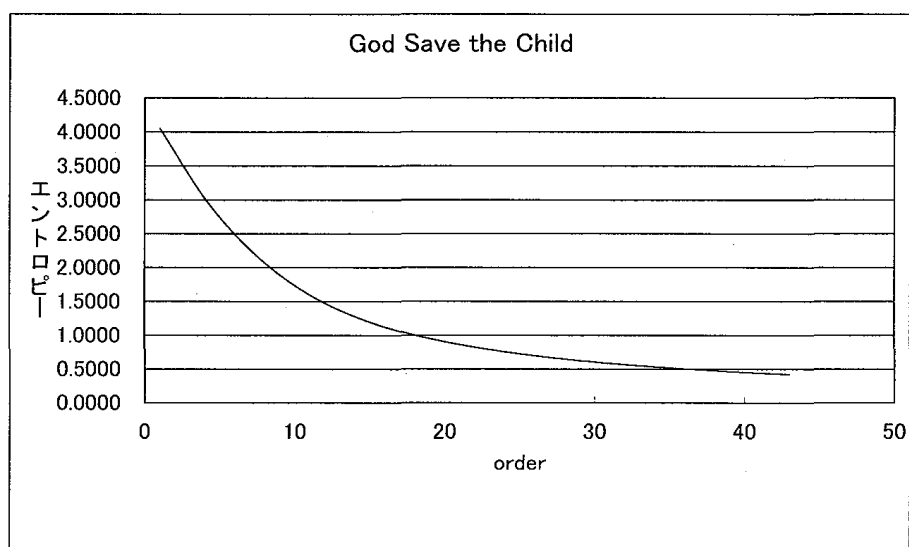
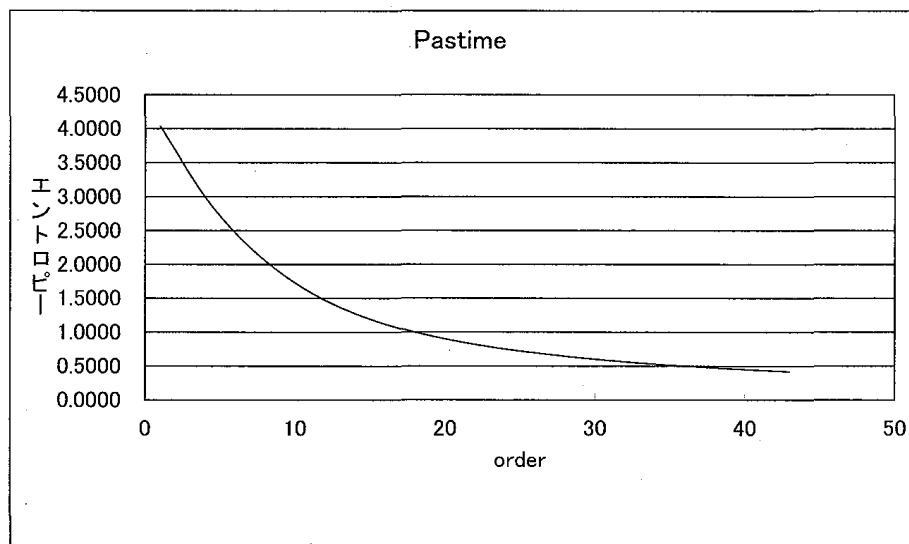
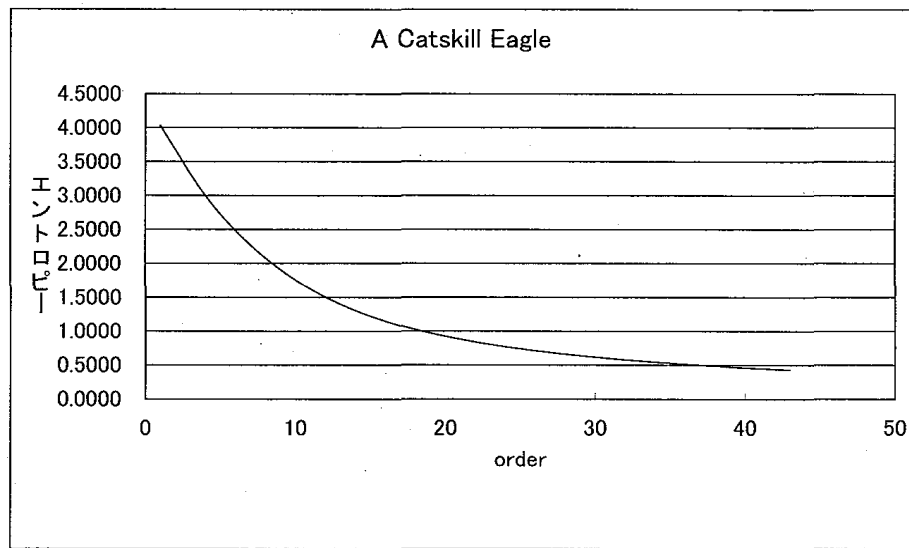
英書 25 冊の第 43 次エントロピー計算



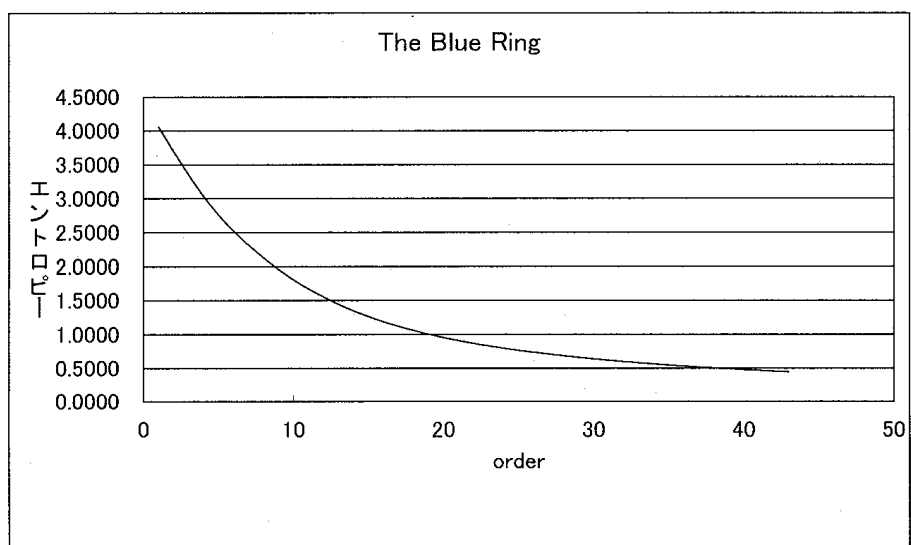
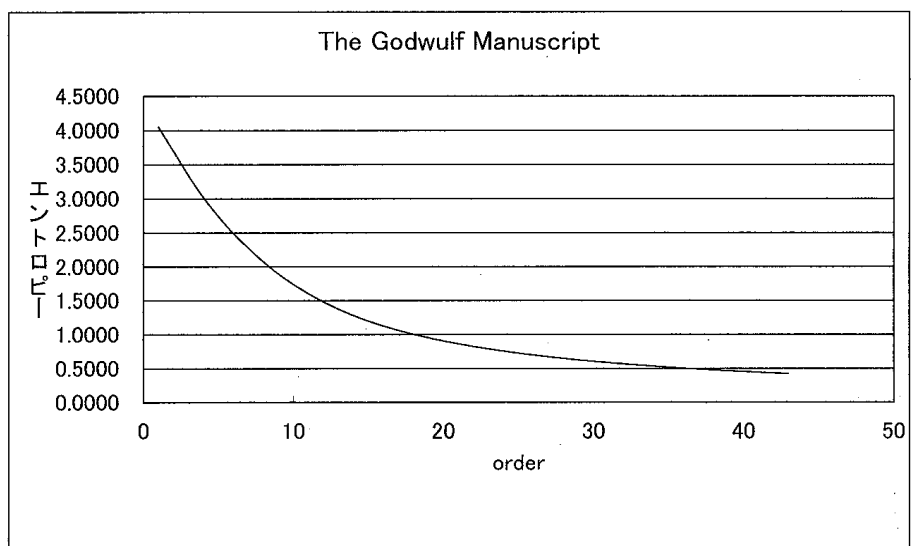
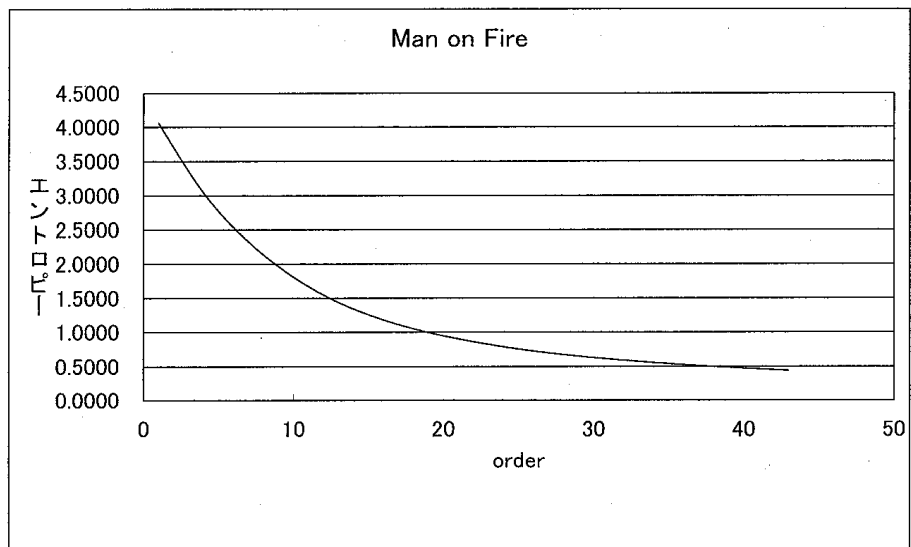


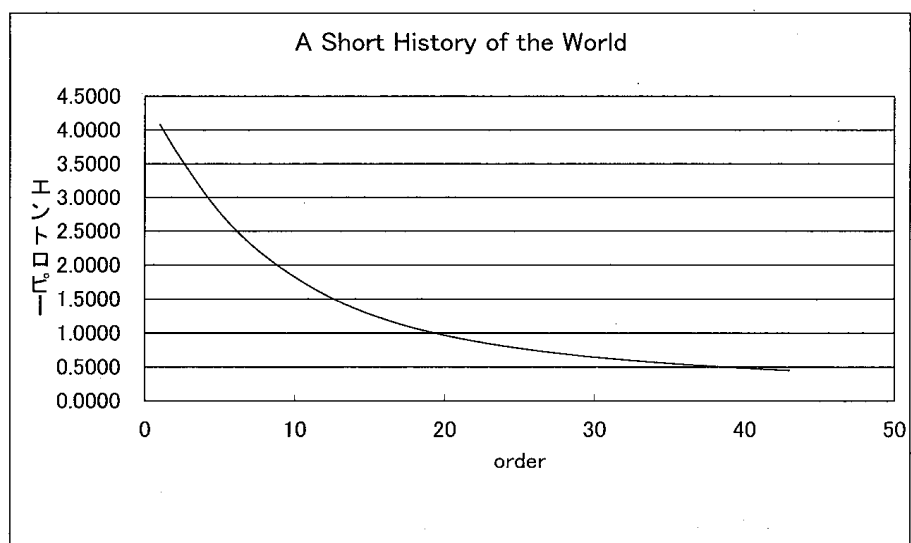
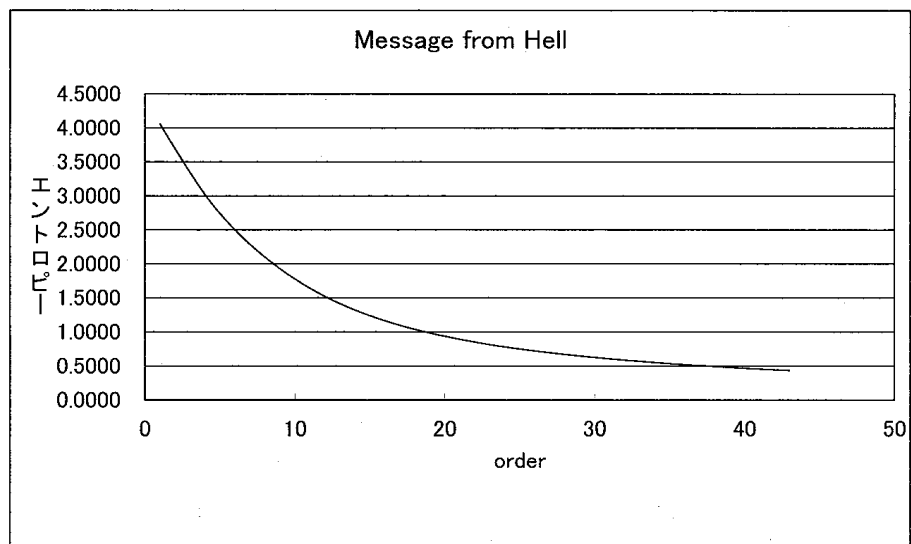
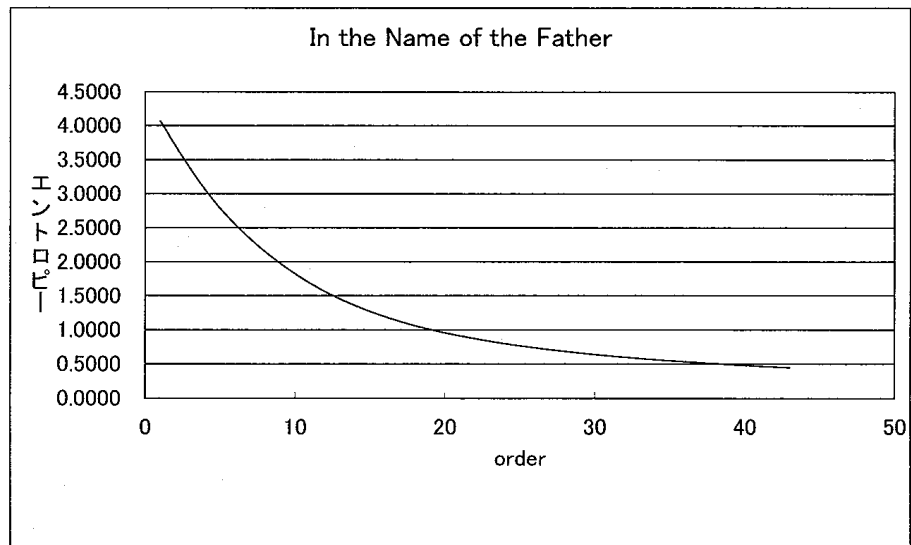
英書 25 冊の第 43 次エントロピー計算



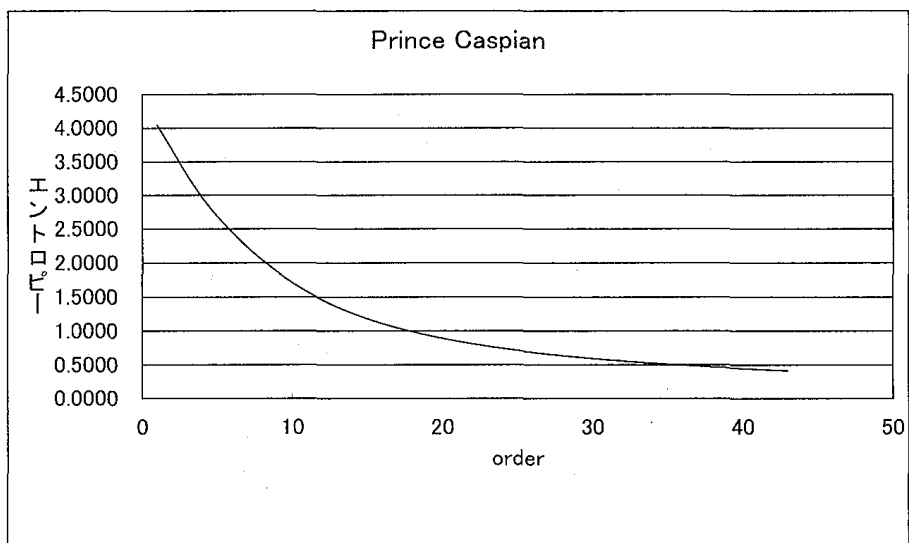
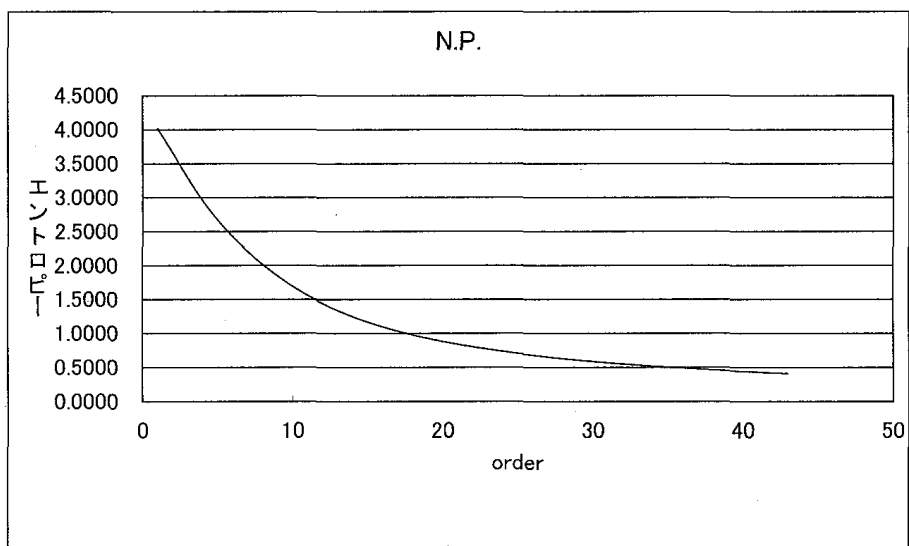
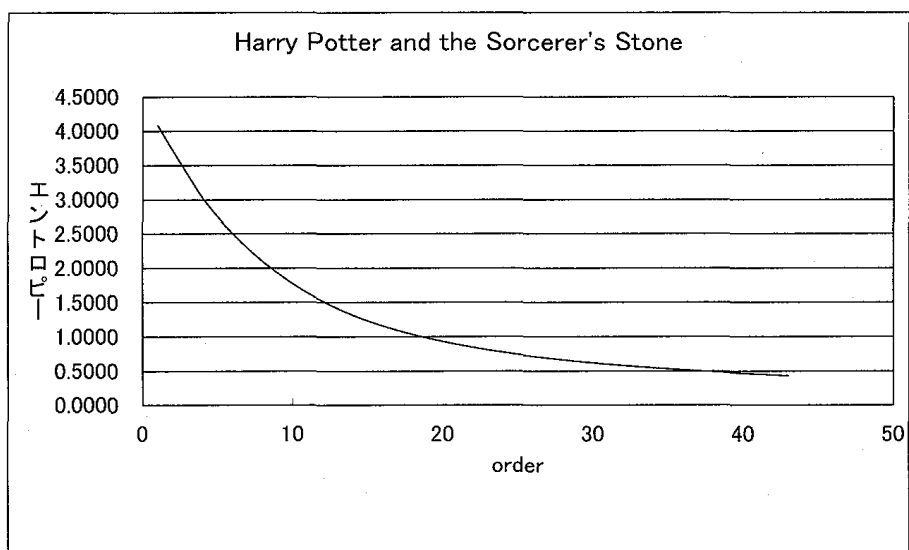


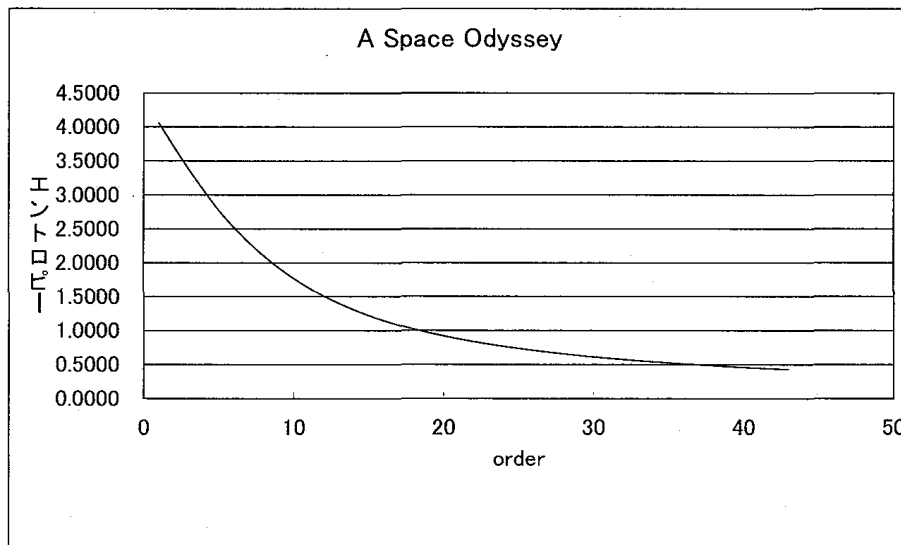
英書 25 冊の第 43 次エントロピー計算





英書 25 冊の第 43 次エントロピー計算



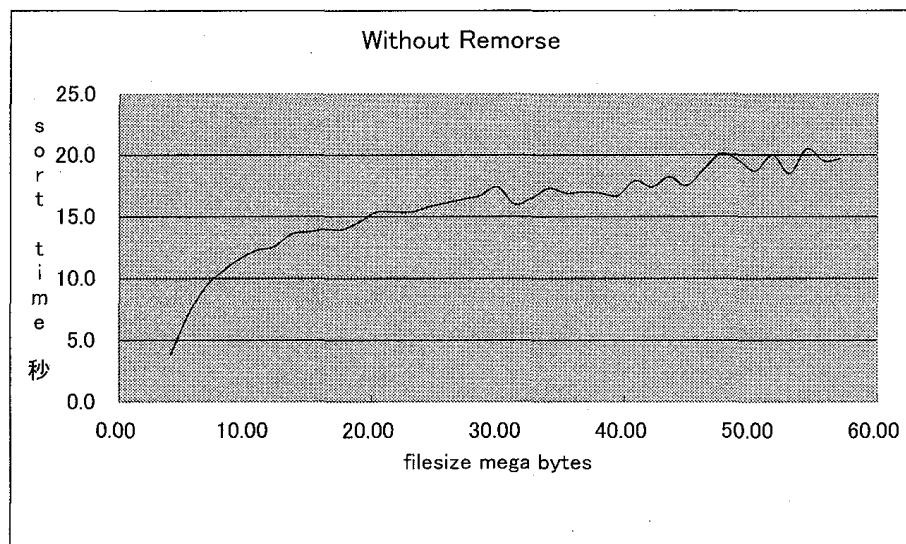


8. sort 時間の興味ある現象

三番目のデータである Tom Clancy の Without Remorse は 1.46 メガバイトの大きさである。本報告が処理した最大のファイルである 1 次エントロピーを計算するためには 1 文字プラス改行キーのレコードを作ると改行キーは 16 進で 0d0a という 2 バイトであるため、3 バイトのレコードが出来上がる。0d はカーソルを最左端に移動させることであり、0a は改行することである。今回はアルファベットとスペースだけ選んだがそれでも 4 メガバイトのファイルになってしまう。Windows2000 の sort を実行したら 2 時間 17 分掛かってしまった。ところが 2 次エントロピーのための sort は 11 分、3 次エントロピーのための sort は 2 分、4 次以上の sort はすべて一分未満であった。経験からファイルが大きくなれば sort 時間も増大すると認識していたため、かなりの違和感を覚えた。

ところが同じ sort を Windows xp で実行すると、今度は逆で、1 次エントロピーのための sort に 3 秒、2 次で 6 秒、と増大し、40 次近くになっても 18 秒から 20 秒の間に飽和するような現象が観察された。ただ、ここから今

回は 43 次エントロピーの計算で終わらせたが、作業時間さえあれば、100 次エントロピーぐらいまでは計算可能のように推測できる。Windows xp による sort 時間の変化をグラフとして次に示す。



7. 結論

- ① 1 次エントロピーはファイル・サイズに関係なく 4.02 から 4.11 までの範囲である。Who killed Cock Robin だけが 4.0 未満なのはアルファベット 27 文字のうち j, q, x, y の 4 文字がないからである。 $p \times \log_2 p$ の数が少なくなるからである。
- ② エントロピーの次数が大きくなるにつれてエントロピーは小さくなるが、その減少の早さはファイルが小さければ早く、大きければ遅い。ファイルが小さければ unique な文字列が少なく、大きければ多い。すなわち小さければ $p \times \log_2 p$ の数が少なく、大きければ、 $p \times \log_2 p$ の数が多いからである。
- ③ 一番大きい Without Remorse で 43 次エントロピーは 0.473839 である。冗長度は 89.55% である。ヤグロム、堀等の予想をはるかに越えている。

参考文献

- (1) アー・エム・ヤグロム, イー・エム・ヤグロム (井関清志・西田俊夫訳): 情報理論入門, みすず書房 (1958) p121
- (2) アー・エム・ヤグロム, イー・エム・ヤグロム (井関清志・西田俊夫訳): 情報理論入門, みすず書房 (1958) p124
- (3) 堀 淳一: エントロピーとは何か, 講談社 (1992) p194
- (4) 南 敏: 情報理論, 産業図書 (1996) p55
- (5) 藤田広一: 情報理論, 昭晃堂 (1996), p24
- (6) 前川 守: 文章を科学する, 岩波書店 (1995) p13
- (7) 前川 守: 文章を科学する, 岩波書店 (1995) p84
- (8) 前川 守: 文章を科学する, 岩波書店 (1995) p48